art news

Kyle J. Knoepfel

art stakeholders meeting

27 April 2017

Detecting multiple events with the same ID

- A recent exception throw was reported on art-users, whose message looked like

```
---- LogicError BEGIN
  Attempt to merge event ranges that both contain one or more of the same events
    SubRun: 1 Event range: [1,51)  vs.
    SubRun: 1 Event range: [1,51)
---- LogicError END
```

- Various ways this can happen—e.g. events with the same `EventID`(s) are included in an art/ROOT file produced via a concatenation job.
- *art* requires that an input file contain unique `EventIDs`—i.e. no duplicates. Art relies on experiment workflows to ensure unique `EventIDs`. Whenever an input-file is read, *art* checks that this precondition is satisfied and throws an exception if it is not.
- There was a suggestion that this checking could be done whenever the file is written/closed and not when it is read.

Event ID checking (1)

- **Current behavior**:
  - When a file is read, and when a given `(Sub)Run` is being prepared for processing, the event IDs associated with that `(Sub)Run` is validated.
  - **Benefit**: Only the relevant `TTree` entries for that `(Sub)Run` are read at a time and not for the whole file.
  - **Drawback**: A file with duplicate events is not detected until it is read.

Event ID checking (2)

- Proposal A: **Check at file close**
  - **Benefit**: The user is notified earlier of an overlap in `EventID`s.
  - **Drawback**: Unless an extra in-memory structure is created, this would require looping through the just-written `TTree`, reading the relevant structures into memory again, and then doing the checking.
  - **Drawback**: This is a check that would be performed per file, which could be expensive and redundant in the context of multiple output streams.

Event ID checking (3)

- Proposal B: **Check before each event processing**
  - Before an event is processed, check that its `EventID` has not been processed already for that job.
  - **Benefit**: Earliest possible check for duplicate events.
  - **Drawback**: Requires creating an in-memory data structure that would be updated and queried for each event processing.

Which problem do we want to solve?

- Proposal A prevents an individual output file from having duplicate events.
- Proposal B prevents a given *art* process from having duplicate events in the stream of files written by each output module.